Supply Chain Risk Management: Enumeration

Animesh Pattanayak
Pacific Northwest National Laboratory
Richland, WA, USA
Animesh.pattanayak@pnnl.gov

Nina Lopez
Pacific Northwest National Laboratory
Richland, WA, USA
Nina.lopez@pnnl.gov

Lucas Tate
Pacific Northwest National Laboratory
Richland, WA, USA
Lucas.tate@pnnl.gov

Travis Anderson
Lawrence Livermore National Laboratory
Livermore, CA, USA
Anderson300@llnl.gov

Abstract *— The systems we use on a day-to-day basis consist of hundreds, if not thousands, of individual components. When considering system security, the inherent risk of each of these components must be considered. A single vulnerability within a component can have a permeating impact throughout the system. Supply chain risks to modern devices can lead to attacks that affect millions of people, as seen in recent years through the Colonial Pipeline attack. To consider the risk posed by individual components, these components must first be identified. Identifying and understanding these risks can minimize damages by providing organizations with methods to reduce risk. One of the first steps toward identifying risks is performing enumeration of the components in a system. While several internet forums mention performing enumeration of hardware systems, no formal or scientific presentation of hardware enumeration processes were found in academic and industry publications. Furthermore, tools for software enumeration do not provide comprehensive insight of the contents of software packages. This paper describes the processes, procedures, and outputs of hardware and software enumeration and proposes a standardized enumeration process that ensures consistency and reproducibility for use in supply chain risk management. The enumeration process proposed creates a more thorough process of data collection and analysis for hardware and software related products. As supply chain security matures, additions to tools and available information could further improve the ability to craft more complete enumerations and analytic conclusions.*

Keywords—component, supply chain risk management, system, critical infrastructure, publicly available information, enumeration, hardware enumeration, software enumeration, bill of materials, software bill of materials, hardware bill of materials

I. Introduction

Traditional supply chain management (SCM) focuses on logistics for maximized efficiency, reliability, and revenue; the newer research area, supply chain risk management (SCRM), additionally aims to identify the risks within a supply chain, such as vulnerabilities, weak points, and break points. In an effort to improve the risk assessment, SCRM aims to look at individual nodes within the supply chain to identify the risk introduced. This in-depth risk assessment may enable an organization to better understand how a product influences its reputation, bottom line goals, and trust in products.

Events in recent years, such as the Colonial Pipeline attack [1], involving supply chain failures have raised awareness of the need for security within supply chains. This priority shift has been recognized by government and industry alike, as single faults within supply chains can have significant negative impact. Efforts to minimize the damages to security, finances, and reputations require identifying and understanding the risks within a supply chain. Because SCRM is relatively new, it is important to understand the taxonomy surrounding the topic and attempt to maintain some standardization. Terms such as elements, processes, and network are used in this context [2]. Though these terms may not always be consistent, the concept of an element, process, and network are maintained. Element refers to the individual components, process refers to the way two or more components may relate to each other, and network defines the grouping of multiple elements and their respective processes [2].

Traditional SCM may indicate where single points of failure exist within a supply chain; consider a critical component with only a single vendor. After identification of these single points of failure, corrective actions can be taken, such as selecting multiple vendors for the component. This is where SCRM builds on SCM.

A single fault in a supply chain can have lasting and noticeable impacts on a business and its product lines. Identifying and understanding various risks can minimize reputation, security, and financial damages by providing organizations within a supply chain with methods to reduce its susceptibilities.

Cyber Testing for Resilient Industrial Control Systems (CyTRICS) [3] is a Department of Energy multi-laboratory effort to increase energy sector cybersecurity and reliability. CyTRICS has strategic partnerships with key stakeholders, including technology developers, manufacturers, asset owners and operators, and interagency partners. CyTRICS has taken initiative on driving the key methodology for enumerating systems. Enumeration is the dissection and documentation of a given device or software program. Enumeration is necessary in SCRM as it enhances accountability, awareness, and anomaly detection by requiring a thorough understanding and analysis of the part being enumerated. Additional research into the product during or after the enumeration process also increases the insight of the product's capabilities and vulnerabilities.

While enumeration of a single component can be useful, developing a consistent and reproducible approach to enumeration enables comparability across enumerations, aggregation across multiple collection teams, and the exchange of enumeration data. By aggregating enumeration data, we can begin to investigate broader questions about supply chains and provide insight into combined attack surfaces. In this paper, efforts to standardize enumerations are presented by detailing the processes, procedures, and outputs to ensure consistency and reproducibility.

When performing an enumeration, it is important to have clear understanding of the scope of enumeration. There may be cases in which most components should be enumerated such as processors, memory chips, and ports. In other cases, chips may be necessary to record but ports are not. One particular delineation with respect to scope is destructive vs. nondestructive enumeration. This scoping indicates whether enumeration should continue when further disassembly of components would potentially cause irreversible damage to the system. One action that may be considered destructive is removing a chip from the circuit board. If an enumeration is requested with the stipulation that the system still be functional upon completion, it is indicative of a nondestructive enumeration.

While the act of performing an enumeration is not inherently illegal, it is strongly suggested that enumeration teams take appropriate measures to understand the licensing information and rights of the hardware or software owner. In short, the enumeration team must ensure that due diligence is practiced before, during, and after the enumeration to ensure appropriate collection methods and data storage.

II. Literature Survey

In preparation of both developing the CyTRICS program and writing this paper, a survey was completed to determine existing practices, theories, and opinions regarding SCRM. As both industry and government continue to place a larger emphasis on securing critical supply chains, there will continue to be a rise in available literature. At this time, the majority of documentation regarding enumeration is specific to software. Less information is available regarding hardware and standardized practices for securing hardware components. Neither software nor hardware have a standardized and documented process that develops a comprehensive view of a given system.

Information regarding supply chain security is also starting to see a larger subset of academic papers. “What do you mean, Supply Chain Security” [2] focuses on defining the framework and taxonomy of supply chain security to better facilitate a common language and set the foundation for standardized practices. This paper defines security in terms of the confidentiality, integrity, and availability triad. Supply chain is divided into elements, processes, and networks. This framework and taxonomy translate to the enumeration process as each step needs to be defined in a reproducible and standardized way.

The literature surrounding enumeration efforts is constantly evolving as enumeration is a somewhat novel process. The enumeration process exists to produce bills of materials (BOMs) for the hardware or software under inspection. Recent articles from OpenBOM, CycloneDX, Fortress, and CISA outline efforts to develop hardware bills of materials (HBOMs) and software bills of materials (SBOMs).

As technology has evolved, the distinction between hardware and software has blurred. Much of modern hardware includes software that runs on it. For example, modern cars may include upward of 100 million lines of code [4]. The OpenBOM article describes the need for a multi-disciplinary BOM to gather and join data together. OpenBOM is a software tool described as a “flexible data model” [4] used to create BOMs.

Another resource for composing BOMs is the CycloneDX project by the Open Web Application Security Project (OWASP). CycloneDX is a model that is capable of bringing together HBOMs, SBOMs, and vulnerability exploitability exchange (VEX) documents, among other data. The CycloneDX BOMs are based on a high-level object model containing data including metadata, components, services, dependencies, compositions, vulnerabilities, and extensions [5].

Once of the primary outcomes of enumerations is the generation of a BOM that provides a detailed view of the exact components that make up a software package or hardware system. The current efforts to generate SBOMs require some form of enumeration, albeit less structured and well defined. By referencing existing SBOMs and efforts to produce SBOMs, the enumeration process can be adjusted to better collect meaningful data.

Part of generating effective SBOMs is to have consistent and standardized terminology and data collected. A well-defined schema can be effective at providing this consistency by specifying the fields to be collected. In addition to stating the terminology, the schema can also define what attributes are to be collected for each of the fields. Efforts to perform enumeration have existed in some capacity prior to being titled enumeration in the form of logical and physical inspection [6] [7], which describe the process for software and hardware enumeration, respectively. Some portions of software enumeration can be automated; however, hardware enumeration can be a more daunting task, particularly for larger systems. For this reason, it may be useful to provide guidance on how to down select the components to be enumerated if resources are not available to perform a full enumeration. Because of their increased potential to be used in an attack, inclusion of all logic-bearing components should be considered a baseline for hardware enumerations [8]. Logic-bearing components include components that possess the capability to perform computation, store data, or communicate with other components, in other words, networking capability.

The National Telecommunications and Information Administration (NTIA) has spearheaded efforts to explore the effectiveness of SBOM within a medical environment [9]. The primary intent of the SBOM is to provide an inventory of the software components and dependencies that make up software systems and define the relationships between the components. Having an inventory provided by the SBOM enables an organization to determine whether certain supply chain concerns will affect their products. Among other benefits, NTIA identifies reduction of cost and reduction of risk as primary benefits. The NTIA SBOM lists the following as baseline information that can be collected for a component:

- Author name
- Component name
- Relationship
- Supplier name
- Version string
- Component hash
- Unique identifier

NTIA performed a proof-of-concept SBOM for medical devices in a healthcare setting to test the efficacy of its SBOM [10]. Through the process, the generation of the SBOM used both manual and semi-automated processes, including scripting languages and software composition analysis tools. The fields identified as important to collect during the proof of concept were as follows:

- Author
- SBOM document name
- List of SBOM components

After completion of the proof of concept, it was determined that the SBOM had cybersecurity benefits across the procurement process, asset management, and enterprise risk management activities. NTIA identified the following strengths, weaknesses, opportunities, and threats associated with the proof of concept.

Strengths included successful access, ingest, parsing, and querying of the SBOM data. The lack of standard format for data, the lack of consistency in naming devices across organizations, and a lack of authoritative sources for certain fields were identified as weaknesses. The proof of concept identified opportunities to create a standard format, use globally unique component identifiers, and begin discussion of including an HBOM. The lack of a defined auditing and validating process presents a threat associated with the data collected as it may not be accurate or complete.

The NTIA, after analyzing use cases, identified the following benefits for three categories of software users: producers, choosers, and operators [11].

- Software producers
- Software choosers
- Software operators

Similar to NTIA, this white paper emphasizes the importance of creating an SBOM as developers often create products that incorporate open-source and commercial software with proprietary code. Identifying and understanding what is in the SBOM is the first step in addressing challenges associated with hardening systems. Using SBOMs allows components to be compared against known databases so vulnerabilities and counterfeit components can be identified more easily.

III. Foundational Data Structures

At the highest level, the object received for enumeration is known as the system, as shown in figure 1. There is only one system per enumeration, which consists of one or more devices. In this framework, a system is an abstract grouping intended to encompass the breadth of the enumeration activity. A device is then comprised of one or more components identified through the enumeration process. For each component, metadata is gathered and stored in isolation. To the extent possible, further decomposition of the component produces more components for which metadata is subsequently captured. This process continues until all components within the system under test have been enumerated. The hierarchical structure of the components as they relate to one another is then described by a set of pairwise relationships where each relationship indicates membership, direction, and nature of the relationship.

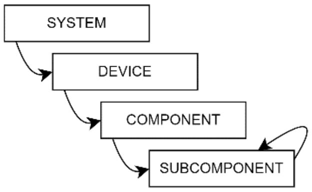

Figure 1: Component Relationship Hierarchy

At the conclusion of an enumeration, what is left is a system containing a list of components that are tied together via relationships. In this way, the nested components of the system and the underlying structure that ties them together are preserved.

IV. Purpose of Enumeration

The purpose of enumeration is to identify all the components within a system. This data then enables us to have a better understanding of how the system works and allows for further research into the system. The components that make up a given product can heavily influence the product's overall exposure to risk based on the component's origin. Understanding how a system works cannot always be assumed by reading the owner's manual, as unintended uses need to be known in critical infrastructure. Enumeration can support security testing by identifying unintended functionality and other security concerns. These unintended uses could have a lasting negative impact on a system, so identifying them is the first step in mitigating any gaps in security. Further research on a system is also needed to best understand all the parts that contribute to the final product. Each component added to a system plays a specific role and can influence the overall security of the system. As a system is typically made up of hardware and software, we have developed enumeration processes specific to both.

A. Awareness and Accountability

Enumeration plays a large role in bringing awareness to both the vendor or manufacturer and the consumer in what actually makes a system and how it can positively or negatively affect their overall network. For hardware this can include memory, processors, ports, and other forms of communication. Software can include types and sizes of files, ports in use, and how the device executes functions, classes, and imports. Each item should be considered a potential entry point for risk and its ability to secure itself and the system from external threats should be scrutinized. Still, inherent risks can occur with a component. Take for example a field programmable grid array, which is intended to be changed post-production. While this allows for flexibility in functionality, it also introduces new risk as it can be manipulated at any time. These types of inherent risks may not be readily known unless all components within a system are cataloged and researched through enumeration. By having awareness of the components that should exist within a system based on manufacturer claims, an organization can hold itself and the manufacturer accountable for verifying the validity of the claims.

B. Documentation

A core outcome of enumerations is to produce an easily readable output of the data collected. The output of the enumerations is a nested directory structure with JavaScript Object Notation (JSON) files that contain the details of the enumerated components as well as any images and documentation associated with the component. In order for the data in the JSON files to be easily searchable and meaningful, it must be standardized and consistent. To ensure this, the data must pass through a validator that compares it to a predefined JSON schema. This is a data structure that describes what is considered acceptable input data. Among other capabilities, the schema defines what field names are valid, the types of those fields, whether the field is required, and the valid values for a given field. After the data in the JSON files have passed the validation step, they are ready to be stored in a data repository. The inclusion of data in a repository enables centralized access to the data in a query-friendly medium.

C. Anomaly Detection

After data collection, documentation, and research have been completed, anomaly detection can more easily be performed. Anomaly detection is identifying what the system is expected to have or do versus what it actually has or does. In some cases, this could mean discovering the system has missing functionality in the software or that counterfeit components have been used in the hardware. It could also mean discovering the system is capable of more than what was intended or that unexpected ports can add further access via communication.

A more serious example of anomalous behavior would be behavior injected from a malicious actor. In December of 2020, FireEye published research of a software supply chain vulnerability being used to insert malicious behavior into the SolarWinds Orion product [12]. Many government agencies were affected by this incident.

A similar technique implemented on systems within critical infrastructure could have much more serious and visible consequences, such as the disruption of critical services as seen in the Colonial Pipeline ransomware attack. Anomaly detection is key to securing critical infrastructure supply chains, so using enumeration to collate data in a standard form is the logical first step in strengthening a system.

V. Process of Enumerations

The enumeration processes for both hardware and software follow an iterative process of unpackaging, research, and documentation that is repeated until the entire product has been enumerated. The processes differ in how the product is unpackaged and documented because of the variance in what data is collected.

A. Hardware Enumeration

1) Unpackaging

When a system is received for enumeration, the first step is to unpackage the system from the shipping container it is received in. The system may be received in a package with supplementary material such as additional cords, power cables, disassembly tools, and documentation. It is important to make note of these supplementary materials as they can be instrumental in proper disassembly or operation of the system. The documentation may provide insight for how certain components need to be removed from the system and in some cases, there are special tools for removing pieces from the unit. After determining which parts of the packaging received are actually part of the system to be enumerated and which parts are supplementary materials, the enumeration process can begin. During the unpackaging process, an inventory should be kept of the parts that arrived in the package and which parts are considered part of the system versus supplementary.

2) Imaging

Imaging includes taking photographs or other visual captures of the system and its components. Imaging should be completed every time a previously undocumented subcomponent is identified during the disassembly phase of enumeration. Images taken should include all critical or relevant information about the component; for example, an integrated circuit where writing on the component is readable should be photographed in a manner that allows the writing to be referenced for information in the future. The time and tools needed for imaging varies based on what is being photographed. For example, capturing an overview image of a system is often as easy as point and shoot, while taking a close-up image of an integrated circuit on a circuit board takes additional stabilization and polarization to optimize the image. Regardless of what is being photographed, the image should be clear and easy to view. An image naming convention is also invaluable when enumerating as recalling specific images needs to be efficient so additional processes can run in a timely fashion.

3) Research

Additional research is almost always required to find key identifying information about a component or system. Identifying information can include part names, part numbers, and the part manufacturer. Research is conducted on the internet using publicly available information (PAI) and results usually include datasheets, manufacturer information releases, and third-party resellers. Discrepancies found in datasheets as compared to the physical component can be critical in anomaly detection.

4) *Documentation*

Data collected through hardware enumeration needs to be documented thoroughly and in a manner that allows for it to be easily ingested and analyzed. A schema, as described in the purpose of enumeration section, gives a way to standardize data so it can be inserted into a repository and ensure all data points are collected. Further information about data structuring related to enumeration is detailed in *Supply Chain Risk Management: Data Structuring* [13].

5) *Disassembly*

While the primary goal is data collection, a good enumeration has a secondary goal of ensuring the system is functional post enumeration. Therefore, disassembly of a device is done in a nondestructive manner that ensures the device can be reassembled and function the same as before disassembly. Each device is disassembled into individual components until doing so would result in the destruction or damage of the device or component. Before beginning disassembly and after each step of the process, documentation and imaging of every new component identified is needed.

6) *Repeat as Necessary*

The previous steps should be repeated recursively until the device has been completely disassembled and all components have been documented. Imaging should also be completed after the system has been reassembled to ensure the system is back in its original condition.

B. *Software Enumeration*

Software enumeration is the process of identifying the individual software files, collecting metadata about the files, and documenting the relationships between them. Because software is not a tangible object like hardware, the enumeration process is inherently different.

1) *Unpackaging*

Generally, software will be provided in a zipped folder. The first step in software enumeration is to handle unzipping the software package as received. These zipped packages may be in a variety of file formats including .zip, .tar, and .7z. There are standard tools that can be used to unzip these packages. After unzipping the package, the tester should have some directory structure with software files. The directory structure will vary with each software package, but generally there will be several levels of nested folders with software files, documentation, and licensing information.

2) *Basic Detail Enumeration*

Software enumeration initially collects details about each file: the file name, hash values, file path, and file type. Collection of these details can be automated using a Python script.

Additional information related to the specific contents of each file can be extracted using knowledge of the specific file format and headers. This can be used to get file type specific metadata such as the vendor, version number of the software, and imported shared libraries. Going a step further with binary files to disassemble/decompile the file can be used to collect information on the functions, classes, strings, and vulnerabilities associated with each file. After running the enumeration script and performing the additional steps listed, a tester should ensure that the fields recorded meet the minimum set of requirements for a complete SBOM as defined by NTIA, described in the Literature Survey.

a) *File Name, File Path, and File Type*

The file name attribute of a file is self-explanatory. The file path attribute is the relative path from the root folder of the software package to the location where the file is located. For basic details enumeration, the file type can be determined based on the extension on the file when an extension is present. For example, a .py file is a Python file and a .docx file is a Microsoft Word file. However, file types can be spoofed by changing the extension to make a file appear as a different type than it actually is.

b) *File Hashes*

The last of the basic details to collect are the hash values associated with a file. Hash values are intended to produce a unique string value based on the contents of the file. A good hashing function will produce a hash such that even a small change in the contents of the file will produce a different hash value. The hash values can be used to verify the identity or ensure the integrity of a file. If the manufacturer claims that the file hash will be *a1b2c3d4* and the computed hash value differs, the tester has reason to believe that some contents of the file have been modified from what the manufacturer released and requires further analysis. There are numerous hashing functions commonly used, but for our enumerations we use SHA1, SHA256, and MD5 because of their current and historic use.

3) *Documentation*

All the details captured must be documented, preferably in a specific format that conforms with the schema mentioned in the earlier section of this paper. By conforming to the schema, data collected can easily be entered into the data repository with limited preprocessing. The last step in software enumeration is to perform any research necessary to ascertain further details about the software package and individual files. The primary goal of this research is to discover known vulnerabilities, datasheets, web pages, and open-source repositories, but any additional information found is useful. The process described should be recursively repeated for each individual file in the directory structure.

VI. TOOLS AND TECHNIQUES

A. *Physical Tools for Enumeration Assistance*

The hardware enumeration process requires multiple physical tools to better deconstruct and document a given device. These tools include the workstation setup, tool kits, cameras, and microscopes. Other physical tools may need to be considered depending on an organization's goals for device enumeration.

1) *Workstation Setup*

The workstation setup will vary depending on the device to be enumerated. Additionally, the number of workers who may be at one table contributes to the size and shape of the desk needed to accommodate people, tools, and the device. After the desk itself has been chosen, all workstations should include antistatic mats to reduce device damages and bodily injuries. Antistatic sheets, often made of foam, should also be used when devices are being deconstructed in order to reduce potential static interactions.

2) *Tool Kit*

Designated tool kits per station are needed for teams planning to work at the same time. The kit should be designed for electronic disassembly and reassembly to best prepare for varying screw heads and sizes, as well as niche tools such as spudgers and angled tweezers. An all-in-one kit can be extremely useful when it comes to organization and storage, but often additional tools such as wrenches may need to be purchased outside of a kit designed specifically for electronics.

3) *Camera*

A camera with the capability to change lenses to take better images is a must. A mirrorless camera is more expensive than a digital single-lens reflex (DSLR) camera and does not have an optical viewfinder. A DSLR camera with a cross-type autofocusing system, tripod mount capabilities, self-cleaning sensor unit, and a minimum of 12 megapixels is recommended for enumerations. The cross-type autofocusing system will ensure that the camera will be in focus, especially when the subject of an image contains vertical lines. This is especially useful when photographing printed circuit boards and other small electronic components. A self-cleaning sensor unit will help ensure that there are minimal dust particles present, which allows for clearer images. Any dust particles on the lens can cause interference and hinder the process of capturing an image of the device. A tripod can be used to help stabilize the camera over the device to allow for a more focused image. By selecting a camera with a minimum of 12 megapixels, the enumeration team can be confident there are enough details captured in the image. If scope and budget allow, a camera with a higher megapixel count should be chosen to enable the enumeration team members to capture additional detail in their images. An 18–55 mm f/3.5–5.6 lens was used for taking images of the component under test. This lens is versatile, allowing for images of the overall device and adequate images of the printed circuit board (PCB). Conformal coating, a protective film placed over integrated circuits and PCBs, can present an issue when taking photographs. Namely, the conformal coating can produce a glare when attempting to capture images of components, causing the text on the component to be unreadable in the image.

4) *Magnification*

Some form of magnification will be required to read identifying text on components, identify anomalies, and count pins, in some cases. Various tools such as microscopes, magnifying glasses, or jeweler's loupes can be used for this purpose. The power of magnification needed depends heavily on the types of components and subcomponents expected to be enumerated. Additionally, lighting and stage options need to be considered as tints of lighting can affect conformal coating differently and stage sizes can impede microscope use. Versatility is also key in identifying key features accurately and efficiently. The team currently uses a jeweler's loupe to move the viewer easily and uses a binocular compound microscope for areas that require a higher magnification. The team is considering more advanced options that can be found in the Conclusions and Future of Work section.

B. *Electronic Tools for Enumeration Assistance*

Enumeration can be a manually intensive task. The hardware systems and software packages received can range from small with fewer than ten components up to very large with hundreds of components. Documenting all of these components, handling the images and datasheets associated, and ensuring the data conforms to the schema and JSON format can be a challenging task. For this reason, eliciting the use of electronic tools can ease the burden on the tester to ensure each of these challenges are met appropriately.

1) *Output Generation*

One electronic tool that can be helpful for ensuring information is formatted correctly is an output generation tool. In this case, the tool may present the user with a frontend web page in which the tester inputs details into appropriately labeled text fields. These values can be used to generate the JSON output file in such a way that there are no concerns about correct formatting. The tool can contain more complexity to represent the relationships between two components but ensure appropriate and consistently formatted output is necessary.

2) *Software Enumeration Scripts*

As discussed in the software enumeration section, a significant portion of the basic software enumeration can be automated with a script to collect the file name, file path, file type, and hashes. This script takes advantage of several Python modules to collect information such as hashes and extension determination. The software enumeration script also outputs to the desired JSON format. The aforementioned script is shown in the Basic Detail Enumeration subsection.

VII. Results and Output

The standardized process of enumeration is still being developed and agreed upon by high levels of government and industry. Due to this, comprehensive examples of enumerations are not publicly available due to the sensitivity of the information.

1) *JSON Structure*

After an enumeration is performed, the data collected and generated needs to be stored in a consistent structure that will make entry into a database seamless. One way to structure data is to use a JSON file format. JSON stores information in key-value pairs in which the key is unique and the value can either be a static value or another JSON object.

For example, the following JSON is valid in that the value associated with the key ***name*** is a static value.

```
{
        “name”: “Alice”
}
```

The next example shows a more complex JSON structure in which the value associated with the key ***person*** is another JSON object.

```
{
        “person”: {
                “name”: “Alice”,
                “age”: 30,
                “profession”: “programmer”
        }
}
```

The data from an enumeration will be stored in a JSON structure, which contains all the pertinent details collected and generated throughout the enumeration process. Each component (hardware), file (software), and relationship will have a separate JSON object in the file. Relationship objects exist to allow a user to understand how two components, or two files are related to each other. For example, if a microprocessor is on a circuit board, the microprocessor is a child of the circuit board. In the case of software, if a file ***hello.c*** exists within the directory ***C_Files***, then the file hello.c is a child of the directory C_Files. By leveraging these relationships, a user is able to represent the structure of a software package or hardware system and understand where within the larger system/package an individual component/file exists.

2) *Proposed Data Schema*

JSON is a useful format for structuring the data, but to achieve consistency, we need to define what fields we expect and what the content of those fields should look like. For this we need to define a JSON schema. The schema allows us to define:

- Fields to be collected
- Human-readable descriptions of the fields
- Examples for data collectors to see
- Expectations for tracking changes
- Logical relationships/dependencies between fields.

Perhaps more importantly, the schema facilitates rapid quality control because it can be used to validate submitted JSON data and point to errors made in collection. This provides a level of assurance that the data submitted from enumeration matches expectations. Enforcing this consistency makes entry into a database automatable. If the JSON conforms to the schema, then each field in the JSON file can be mapped to a field in a database.

3) *Repository Motivation and Objectives*

Repositories can come in many forms and are generally shaped by the nature of the data that will be stored in them. Some examples of potential repository forms include databases (ex: MySQL, Postgres, NoSQL, MariaDB). While specifics of the storage are outside of the scope of this document, the primary motivations for the repository are important to understand as part of the complete process.

The primary objectives of the repository are to archive the data and perhaps more importantly to make it accessible for reuse. Using the well-structured data collection can facilitate direct ingestion into a database. From there it is important to consider how the data will be indexed for search and to ensure that the structure you choose aligns well with the anticipated research. Answering the following questions at the onset will help ensure that the resulting data is well aligned with the intended use.

- Is string searching sufficient?
- Do you need to traverse relationships in the query?

VIII. Conclusions and Future Work

Developing and integrating a standard enumeration process is essential to securing critical infrastructure and the associated supply chains. The proposed enumeration is well formatted to service hardware and software related products to create a more thorough method of data collection and analysis. Current methods are sufficient but additions to accessible tools and available information would better situate teams in crafting complete enumerations, and ultimately, analytic conclusions.

While enumeration is an integral step in understanding the risk associated with a system, it is only one of the first steps in completing a full risk analysis. Enumeration informs the tester of the system components but does not provide any risk level indicator. Some current challenges and limitations being experienced by those developing and implementing enumeration procedures include differences in terminology and catering to varying levels of expertise and tools available. In order to have a longstanding and successful methodology, terminology must be standardized across industry so cross pollination of information and ideas can occur. Additionally, this standardized process needs to be scalable for complexity of the device as well as the existing capabilities an enumeration team may have. After enumeration, performing PAI research on each component will inform the tester of any known and documented susceptibilities associated with the component that influences the overall risk. In addition to what is publicly available, performing hands on analysis can assist in discovering new risk indicators. When enumeration is performed in conjunction with PAI research and analysis, a more complete risk profile can be generated for the system.

Future work should include the funding for an enumeration on a piece of obsolete hardware and software to thoroughly demonstrate and document the process.

A. *Tools to Consider*

1) *Handheld Microscope*

Adding a handheld microscope to physical tools could prove to be beneficial in not only imaging but also documentation. There are a wide variety of handheld microscopes on the market with differing specifications based on intended use. A handheld microscope with dynamic range, polarization, large working distance, and a wide-angle lens will allow closer and clearer imaging than a camera with an 18–55 mm lens. Additionally, the versatility of a handheld device allows the user to angle the microscope in a manner that a traditional microscope would be unable to reach.

2) Digital Microscope

Still, there are more advanced digital microscopes that can be used to construct virtual three-dimensional representations of the product as well as up to 6,000 times magnification. Determining which microscope, if any, will enhance current enumeration procedures will depend on cost, the ability of team members to learn how to use, how the added features influence data collection, and any added security risks.

3) Additional Lens and Lighting Options

A 100 mm f/2.8 macro lens would be a great additional lens. A macro lens would enable capturing high-quality images of the text on small components, which are traditionally hard to read by the human eye. Additionally, having multiple LED lamps or overhead lights would allow for better images by providing proper lighting from several angles without emitting much heat. An LED light and the macro lens would allow taking quality images of the PCB with minimal problems from conformal coating.

4) Vendor/Manufacturer Supplied BOMs

Detailed BOMs from vendors would help confirm the contents of a device and make finding discrepancies easier. Additionally, having access to such information could allow enumeration teams to have varying levels of disassembly performed depending on the client’s needs. For example, confirming parts are present by comparing to the BOM from the vendor or manufacturer would be a faster process than discovering, identifying, and confirming a component is supposed to be in the system. The time spent on enumeration would depend heavily on the type of information already provided and also would influence what types of tools would be necessary for the type of enumeration.

Acknowledgment

The authors would like to thank their partners throughout the U.S. Department of Energy, Office of Cybersecurity, Energy Security, and Emergency Response (DOE CSESER)'s CyTRICS program, including Idaho National Laboratory, Lawrence Livermore National Laboratory, Oak Ridge National Laboratory, Pacific Northwest National Laboratory, and Sandia National Laboratories.

References

[1] G. M. Makrakis, C. Kolias, G. Kambourakis, C. Rieger and J. Benjamin, "Industrial and Critical Infrastructure Security: Technical Analysis of Real-Life Security Incidents," *IEEE Access*, vol. 9, 2021.

[2] J. Smith and J. Teuton, "What do you mean, Supply Chain Security? A Taxonomy and Framework for Knowledge Sharing," in *Hawaii International Conference on System Sciences*, Waikoloa Village, 2017.

[3] Idaho National Laboratory, "CyTRICS: Cyber Testing for Resilient Industrial Control Systems," U.S. Department of Energy: Office of Cybersecurity, Energy Security, and Emergency Response, [Online]. Available: https://cytrics.inl.gov/. [Accessed 05 01 2023].

[4] O. Shilovitsky, "OpenBOM," 27 05 2021. [Online]. Available: https://www.openbom.com/blog/bill-of-materials-lifecycle-software-vs-hardware-and-multi-disciplinary-bom. [Accessed 10 01 2023].

[5] CycloneDX, "Hardware Bill of Materials," OWASP, [Online]. Available: https://cyclonedx.org/capabilities/hbom/. [Accessed 10 01 2023].

[6] N. Lopez, E. Pollans, M. Kirkland, J. Seaman, A. Pattanayak and L. Tate, "Landscape Analysis: Supply Chain Risk Management Established Methodologies," Unpublished., 2022.

[7] S. Pal, "Technology for Reducing Risks of Equipment Failures and Cyber Incident," New Orleans, 2022.

[8] N. Lopez and E. Pollans, "Site Best Practices," Unpublished, 2023.

[9] National Telecommunications and Information Administration, "SBOM at a Glance," 27 April 2021. [Online]. Available: https://www.ntia.gov/files/ntia/publications/sbom_at_a_glance_apr2021.pdf. [Accessed 9 January 2023].

[10] National Telecommunications and Information Administration, "Software Component Transparency: Healthcare Proof of Concept Report," 1 October 2019. [Online]. Available: https://www.ntia.gov/files/ntia/publications/ntia_sbom_healthcare_poc_report_2019_1001.pdf. [Accessed 9 January 2023].

[11] National Telecommunications and Information Administration, "Roles and Benefits for SBOM Across the Supply Chain," 8 November 2019. [Online]. Available: https://www.ntia.gov/files/ntia/publications/ntia_sbom_use_cases_roles_benefits-nov2019.pdf. [Accessed 9 January 2023].

[12] C. Jaikaran, "SolarWinds Attack—No Easy Fix," Library of Congress, Washington, D.C, 2021.

[13] N. Lopez, A. Pattanayak and J. Smith, "Supply Chain Risk Management: Data Structuring," in *2022 Resilience Week* , National Harbor, 2022.